Electroniques des composants et systèmes

Chapitre 03 :

Le microprocesseur

Qu'est ce qu'un CPU?

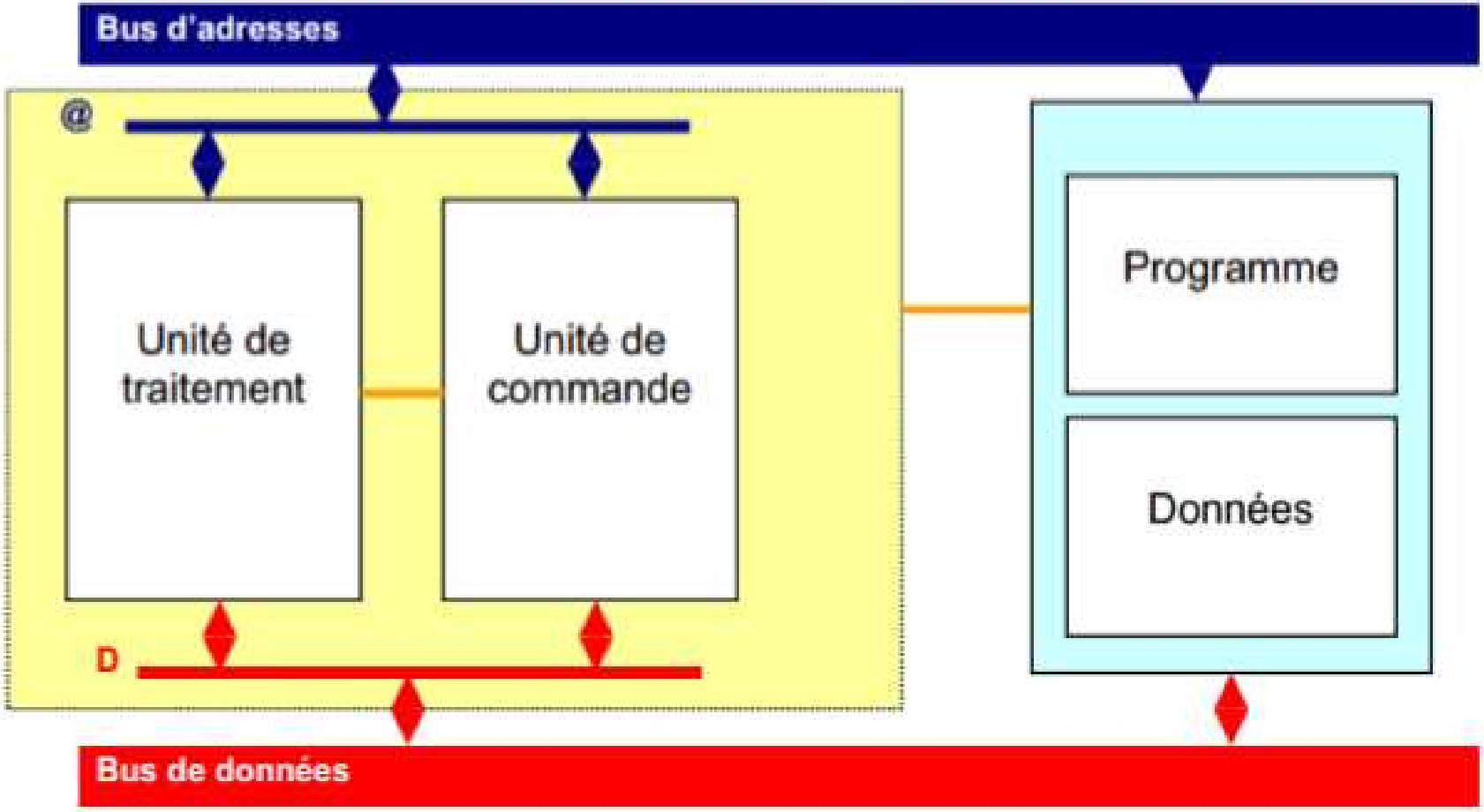
- Un microprocesseur est un circuit intégré complexe représentant le cerveau de l'ordinateur,
- Il résulte de l'intégration sur une puce de plusieurs circuits logiques combinatoires)fonctions logiques et arithmétiques) et séquentielles)registres, compteur, etc.(…
- A l'heure actuelle, un microprocesseur regroupe sur quelques millimètre carré des fonctionnalités toujours plus complexes.
- le CPU est chargé d'interpréter et d'exécuter les instructions d'un programme, de lire ou sauvegarder des données dans des mémoires et de communiquer avec les unités d'échange.
- Toutes les activités du microprocesseur sont cadencées par une horloge.

Architecture d'un CPU

Un CPU est construit autour des éléments principaux suivants:

- Une unité de commande,
- Une unité de traitement,
- Des registres chargées de stocker les différentes informations à traiter,
- Ces trois éléments sont reliés entre eux par des bus internes permettant les échanges d'informations.

Architecture d'un CPU



Unité de commande

L'unité de commande permet :

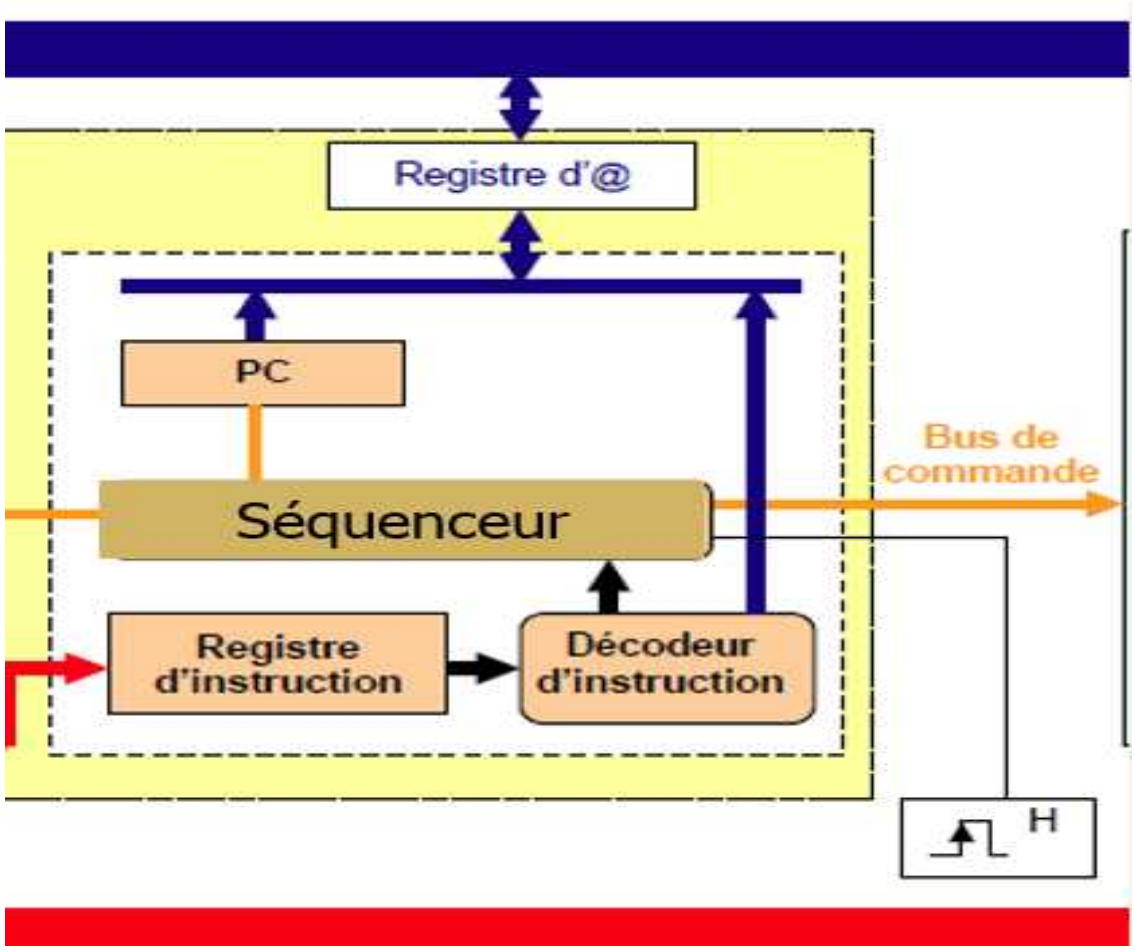
- de séquencer le déroulement de l'exécution des instructions,
- d'assurer le décodage de l'instruction (qui est codée sous forme binaire(
- d'organiser l'exécution de l'instruction,
- d'effectuer la préparation de l'instruction suivante.

Unité de commande

L'unité de commande est composée par:

- le compteur ordinal ou de programme (CO ou PC) est un registre qui contient l'adresse de l'instruction à exécuter. Il initialisé avec l'adresse de la première instruction du programme.
- le registre d'instruction (RI): qui contient l'instruction à exécuter,
- le décodeur d'instruction :qui décode l'instruction qui est sous forme binaire en identifiant l'opération et les opérandes.
- Le bloc logique de commande (ou séquenceur) : est un automate (circuit) qui organise l'exécution des instructions au rythme d'une horloge en élaborant des ordres de commandes aux différents composants internes (du cpu (ou externe (de l'extérieur du CPU : MC(

Unité de commande



Unité de traitement

- C’est le cœur du CPU. Elle regroupe les circuits qui assurent les traitements nécessaires à l'exécution des instructions
- Elle est composée essentiellement de :
  - l'Unité Arithmétique et Logique)UAL) qui est un circuit complexe assurant:
    - les fonctions logiques)ET, OU, Comparaison, Décalage ,etc(… et,
    - les fonctions arithmétiques)Addition, soustraction.(……
  - Le registre d'états (PSW, Processor Status Word), permet de stocker des indicateurs sur l'état du système (retenue, dépassement, etc.)
  - Les accumulateurs sont des registres de travail qui servent à stocker une opérande au début d'une opération arithmétique et le résultat à la fin de l'opération.

Unité de traitement

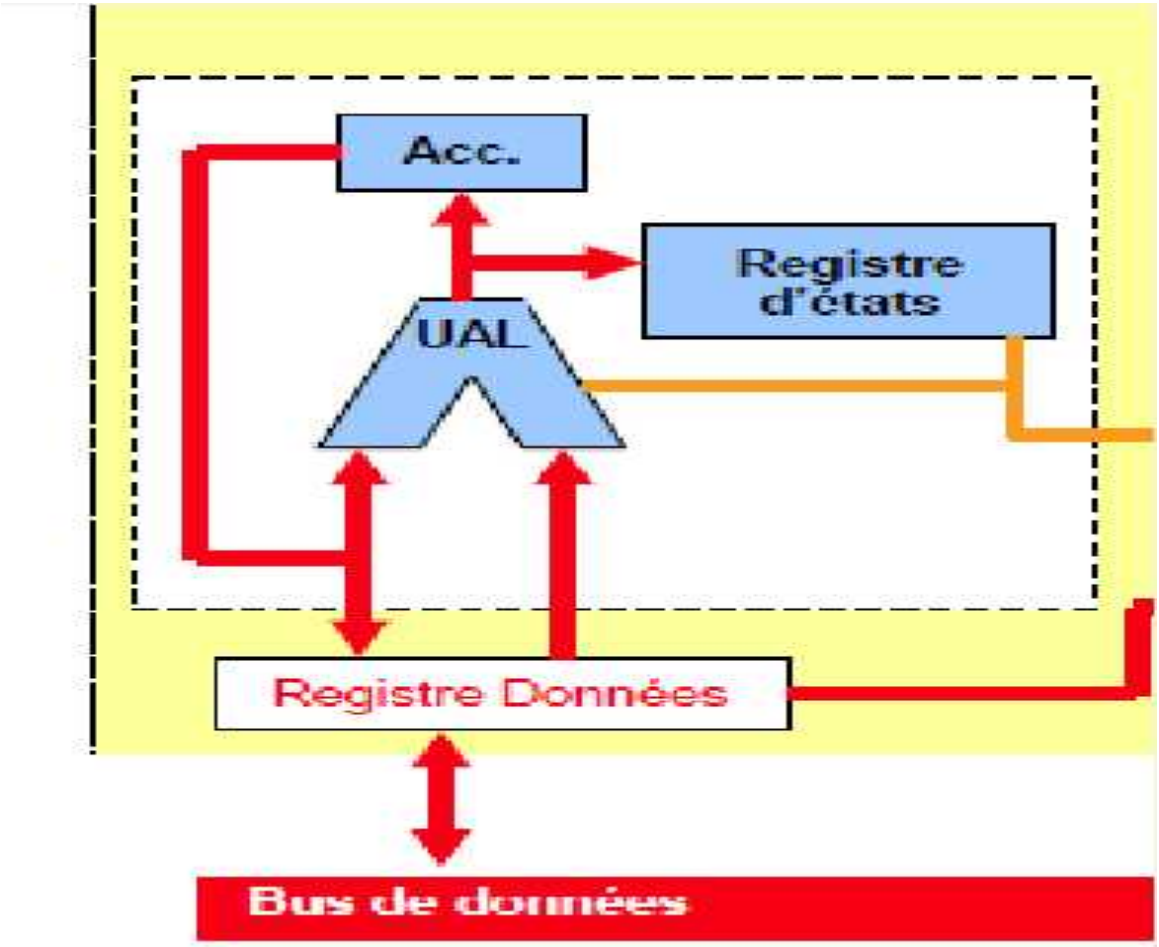
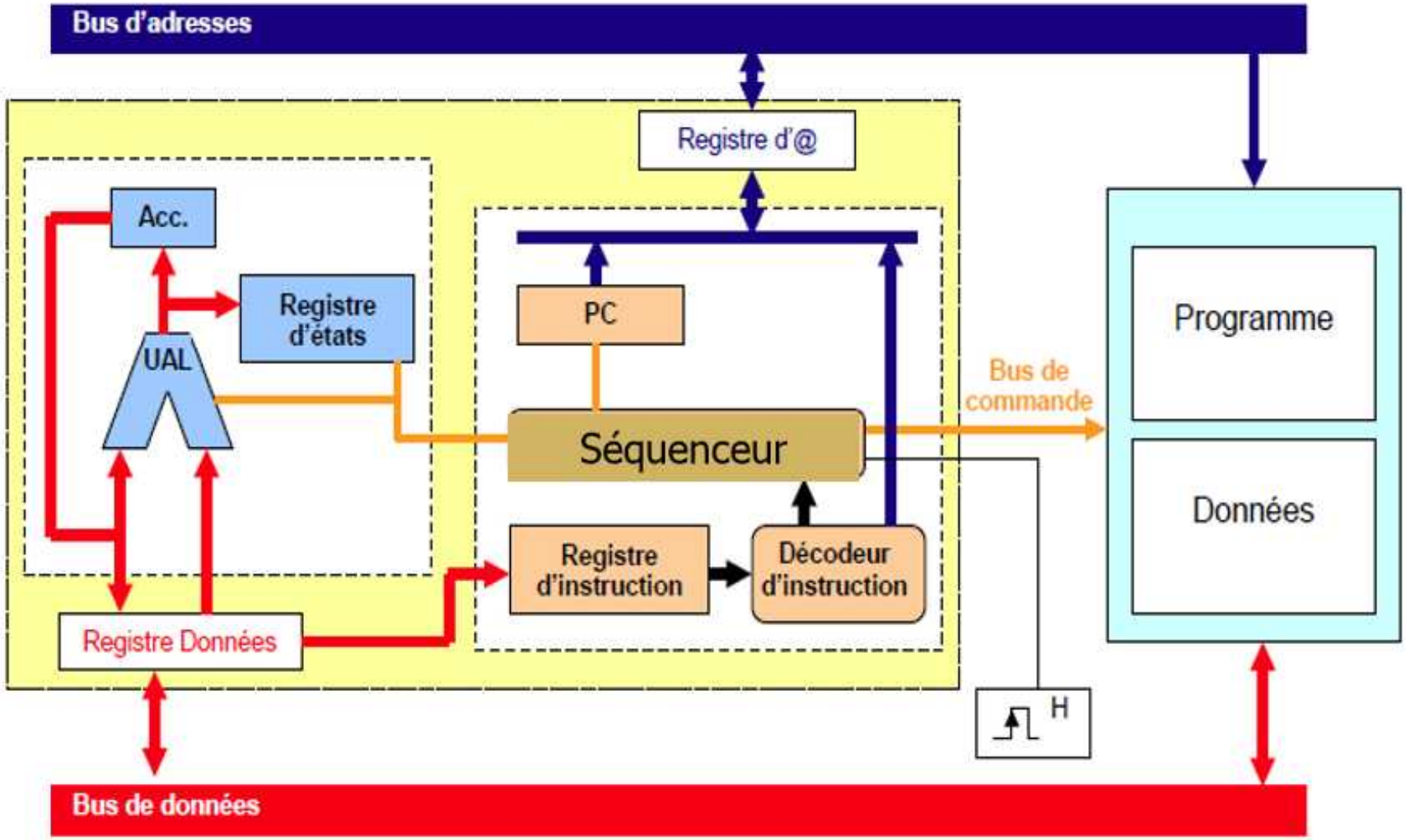


Schéma fonctionnel



Cycle d’exécution d’une instruction

Le traitement d’une instruction peut être décomposé en trois phases.

- Phase :1Recherche de l'instruction à traiter
  - Le PC contient l'adresse de l'instruction à exécuter. Cette valeur est copiée dans le registre d'adresses par le séquenceur qui émet un ordre de lecture.
  - Au bout d'un certain temps (temps d'accès à la mémoire), le contenu de la case mémoire sélectionnée est disponible sur le registre de données.
  - L'instruction est stockée dans le registre instruction du processeur.

Cycle d'exécution d'une instruction

- Phase : 2 Décodage de l'instruction et recherche de l'opérande
  - Le registre d'instruction contient l'instruction codée en binaire
  - Le décodeur décode cette instruction en identifiant l'opération et les opérandes
  - Le séquenceur donne l'ordre à L'UAL de charger le circuit réalisant l'opération en cours
  - Si l'instruction nécessite une donnée en provenance de la mémoire, l'unité de commande récupère sa valeur sur le bus de données.
  - L'opérande est stockée dans un registre.

Cycle d'exécution d'une instruction

- Phase :3 Exécution de l'instruction
  - Le circuit relatif à l'opération à réaliser, dans l'UAL, exécute l'opération (instruction.(
  - Le résultat de l'opération (s'il existe) est stocké dans ACC puis dans le Registre données puis sauvegardé en MC
  - L'unité de commande (séquenceur) positionne le PC pour l'instruction suivante.

Jeu d'instructions

- La première étape de la conception d'un microprocesseur est la définition de son jeu d'instructions.
- Le jeu d'instructions décrit l'ensemble des opérations élémentaires que le microprocesseur pourra exécuter.
- Il va donc en partie déterminer l'architecture du microprocesseur à réaliser et notamment celle du séquenceur.
- A un même jeu d'instructions peut correspondre un grand nombre d'implémentations différentes du microprocesseur.

Instruction niveau machine

- Une instruction est l'opération élémentaire que le processeur peut accomplir.
- Une instruction est composée de plusieurs champs:
  - le code opération « instruction »: code de l'action que le processeur doit exécuter ;
  - le code opérande, c'est les paramètres de l'action qui peuvent être une donnée ou bien une adresse mémoire).

Code instruction	Code opérande
1001 0011	0011 1110

- On parle d'instruction à n adresses si elle possède n champs opérandes (n = 0,1,2,3,4 au plus)

Instruction niveau machine

- Le nombre d'instructions du jeu d'instructions du CPU est directement lié au format du code opération (nombre de bits).
  - Ainsi un code opération sur un octet permet de distinguer au maximum 256instructions différentes dans le jeu d'instructions.
- Chaque instruction nécessite un certain nombre de cycles d'horloges pour s'effectuer; Le nombre de cycles dépend de la complexité de l'instruction.
  - A chaque top d'horloge le processeur exécute une action, correspondant à une instruction ou une partie d'instruction.

Les types d'instruction

Les instructions que l'on retrouve dans le jeu d'instructions du microprocesseur peuvent être classées en 4groupes:

- Transfert de données pour charger ou sauvegarder les données en mémoire, effectuer des transferts de registre à registre, etc…
- Opérations arithmétiques : addition, soustraction, division, multiplication
- Opérations logiques : ET, OU, NON, NAND, comparaison, test, etc…
- Contrôle de séquence : branchement, test, etc…

Caractéristiques du CPU

- Le jeu d'instructions qu'il peut exécuter.
- La complexité de son architecture. Cette complexité se mesure par le nombre de transistors contenus dans le CPU. (Plus le microprocesseur contient de transistors, plus il pourra effectuer des opérations complexes, et/ou traiter des chiffres de grande taille.(
- La Largeur des données : le nombre de bits que le processeur peut traiter ensemble. Les CPU actuels peuvent traiter des nombres sur 64 bits.

Caractéristiques du CPU

- La finesse de gravure (nm) : le diamètre en nanomètre du plus petit fil reliant deux composantes du CPU.
  - En 2014 on arrive à des finesses de gravure de l'ordre de 10 nm.
- La vitesse de l'horloge. plus la vitesse augmente, plus le processeur effectue d'instructions en une seconde.
- Les Performances d'un microprocesseur : MIPS et CPI

Performance d'un CPU

On peut caractériser la puissance d'un microprocesseur par le nombre d'instructions qu'il est capable de traiter par seconde. Pour cela, on définit:

- le CPI (Cycle Par Instruction) : qui représente le nombre moyen de cycles d'horloge nécessaire pour l'exécution d'une instruction pour un microprocesseur donné.
- le MIPS (Millions d'Instructions Par Seconde): qui représente la puissance de traitement du microprocesseur.

$$MIPS = \frac{F_H}{CPI}$$

Performance d’un CPU

- Pour augmenter les performances d’un microprocesseur, on peut donc:
  - soit augmenter la fréquence d'horloge)limitation matérielle,(
  - soit diminuer le CPI)choix d'un jeu d'instruction adapté.(
  - soit améliorer l’architecture de base)amélioration technologique(
- Le rôle de l’horloge est de cadencer le rythme du travail du processeur. + la vitesse augmente, + le processeur execute d'instructions en une seconde.
- Inconvénients de l'augmentation de la fréquence :
  - le processeur consomme d'électricité,
  - il se chauffe ce qui nécessite une solution de refroidissement du processeur adaptée ;

Architecture CISC vs RISC

Il existe deux catégories d'architectures pour le microprocesseurs :

- L'architecture CISC)Complex Instruction Set Computer(basée sur un jeu d'instructions complexes
- L'architecture RISC (Reduced Instruction Set Computer(basée sur un jeu d'instructions élémentaires (simples ou réduits(

Architecture CISC

- vue que la mémoire travaillait très lentement par rapport au processeur, il était plus intéressant de soumettre au microprocesseur des instructions complexes (minimiser l'accès à la mémoire(
- On a donc eu tendance à incorporer au niveau processeur des instructions plus proches de la structure des langage de haut niveau

Architecture CISC

- CISC est une architecture où le microprocesseur exécute des tâches complexes par instruction unique qui nécessite un plus grand nombre de cycles d'horloge
- Le code machine de ces instructions varie d'une instruction à l'autre et nécessite donc un décodeur complexe (micro-code) pour les éxecuter

Architecture RISC

- Des études statistiques ont montré que les programmes se contentaient le plus souvent d'affectations et opérations arithmétiques simples.
- Ainsi, %80 des traitements des langages de haut niveau faisaient appel à seulement 20% des instructions du microprocesseur.
- D'où l'idée de réduire le jeu d' instructions à celles le plus couramment utilisées et d'en améliorer la vitesse de traitement.

Architecture RISC

- RISC est une architecture dans laquelle les instructions sont en nombre réduit)chargement, branchement, appel sous-programme.(
- Chacune de ces instructions s'exécutent ainsi en un cycle d'horloge.
- Les accès à la mémoire s'effectue seulement à partir de deux instructions (Exemple: Load et Store).
- Remarque :par contre, les instructions complexes doivent être réalisées à partir de séquences basées sur les instructions élémentaires, ce qui nécessite un compilateur très évolué dans le cas de programmation en langage de haut niveau.

Comparaison CISC vs RISC

- Si on diminue le nombre d'instructions, on crée des instructions complexes (CISC(qui nécessitent plus de cycles pour être décodées
- Si on diminue le nombre de cycles par instruction, on crée des instructions simples)RISC(mais on augmente alors le nombre d'instructions nécessaires pour réaliser le même traitement.

Comparaison CISC vs RISC

Architecture RISC	Architecture CISC
1. instructions simples ne prenant qu'un seul cycle 2. instructions au format fixe 3. décodeur simple (câblé) 4. beaucoup de registres 5. seules les instructions LOAD et STORE ont accès à la mémoire 6. peu de modes d'adressage 7. compilateur complexe	1. instructions complexes prenant plusieurs cycles 2. instructions au format variable 3. décodeur complexe (microcode) 4. peu de registres 5. toutes les instructions sont susceptibles d'accéder à la mémoire 6. beaucoup de modes d'adressage 7. compilateur simple

Comparaison CISC vs RISC

- Aujourd'hui le « pur » RISC à évolué on trouve dans ces architectures des instructions complexes
- On trouve des architectures mixtes à noyau RISC et extension CISC
- L'évolution des technologies définira et imposera des compromis permettant d'aller vers des machines plus efficaces

Le parallélisme

D’autres approches, améliorant les performances d’un microprocesseur, sont possibles, elles portent sur la parallélisation des opérations (faire plusieurs choses à la fois:(

- Au niveau des instructions
- Au niveau des processeurs

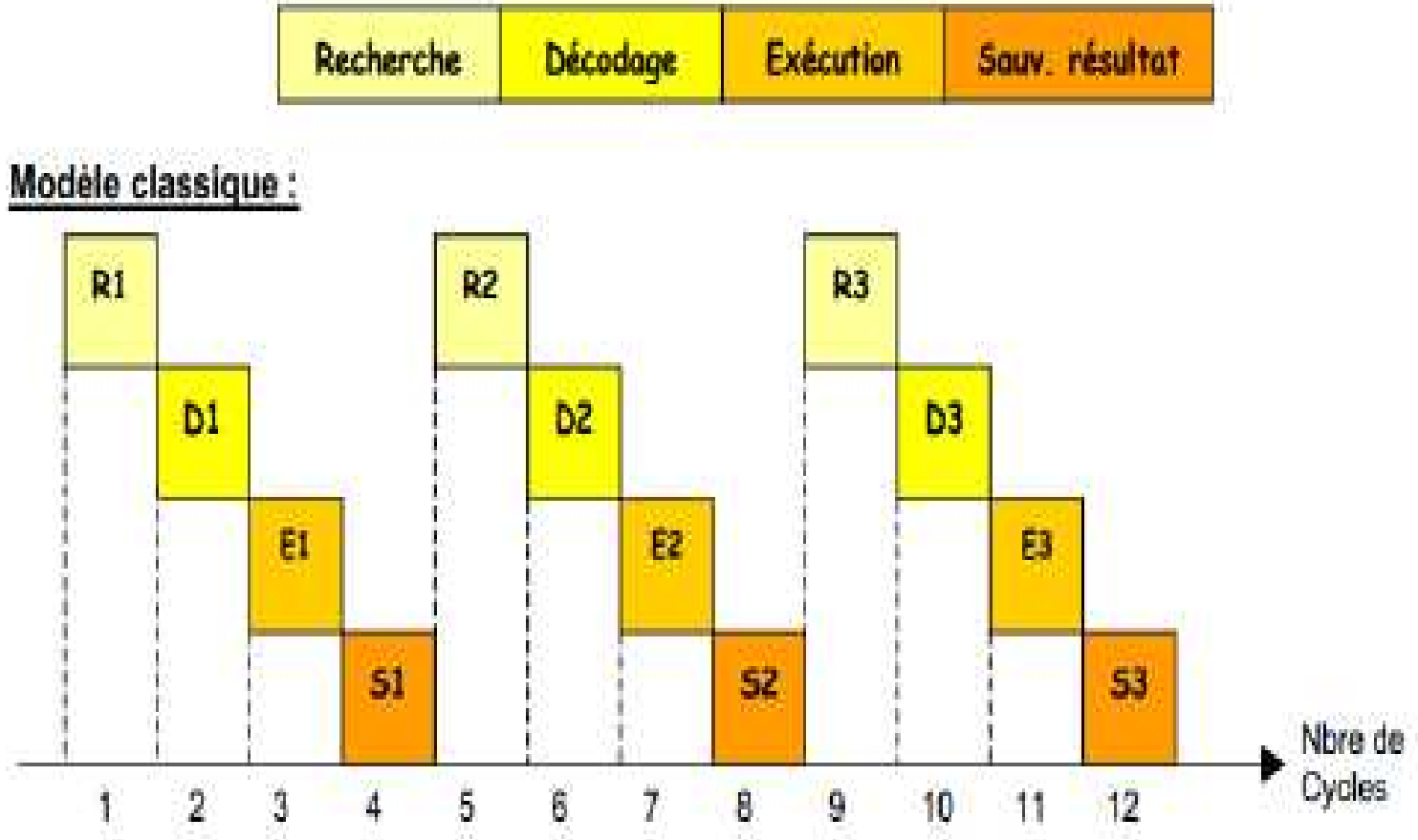
Le parallélisme des instructions

- Un programme informatique est un ensemble d'instructions exécutées par un processeur.
- Chaque instruction nécessite un à plusieurs cycles d'horloge,
- Le microprocesseur séquentiel exécute l'instruction suivante lorsqu'il a terminé l'instruction en cours. Donc, si l'instruction demande la lecture des variables depuis la MC, le CPU ne fait rien dans ce temps d'attente des variables. Pour remédier à ce problème de chômage de CPU, une solution est de paralléliser l'exécution des instructions
- Dans le cas du parallélisme d'instructions, le microprocesseur pourra traiter plusieurs instructions dans le même cycle d'horloge, à condition de ne pas utiliser simultanément une même ressource interne)variables, registres.(..... ,
- Autrement dit, le processeur exécute, en parallèle, des instructions qui se suivent, et ne sont pas dépendantes l'une de l'autre.

Le parallélisme des instructions

- **Principe** : l'exécution d'une instruction est décomposée en une succession d'étapes et chaque étape correspond à l'utilisation d'une des fonctions du microprocesseur.
- Le cycle d'exécution d'une instruction est composée de 4 étapes:
  1. Recherche de l'instruction (PC, R@, MC, RD, RI)
  2. décodage de l'instruction (décodeur d'instruction)
  3. Exécution (unité de traitement)
  4. sauvegarde des résultats
- Lorsqu'une instruction se trouve dans l'une des étapes, les composants associés aux autres étapes ne sont pas utilisés.

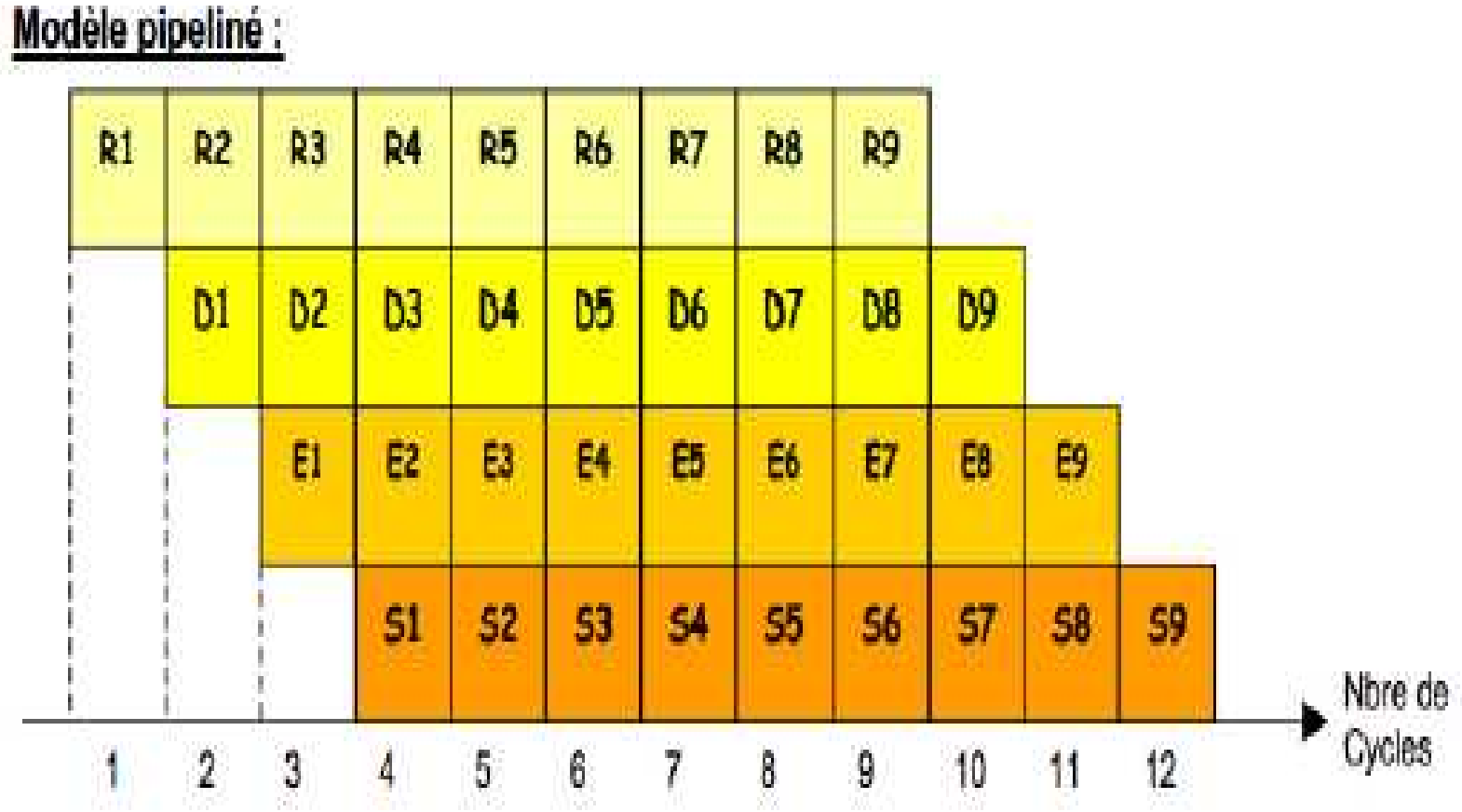
Exemple du modèle classique)CPU séquentiel(



Parallélisme des instruction (architecture pipeline)

- Le fonctionnement d'un microprocesseur simple (séquentiel) n'est donc pas efficace.
- L'architecture pipeline)en file(permet d'améliorer l'efficacité du microprocesseur.
- **Principe:** lorsque la première étape de l'exécution d'une instruction est achevée, l'instruction entre dans la seconde étape de son exécution et la première ètape de l'exécution de l'instruction suivante débute.

Exemple du modèle pipeliné



Parallélisme des instruction)architecture pipeline(

- Il peut donc y avoir une instruction en cours d'exécution dans chacune des étapes et chacun des composants du microprocesseur peut être utilisé à chaque cycle d'horloge.
- L'efficacité est maximale. Le temps d'exécution d'une instruction n'est pas réduit mais le débit d'exécution des instructions est considérablement augmenté.
- Une machine pipeline se caractérise par le nombre d'étapes utilisées pour l'exécution d'une instruction, on appelle aussi ce nombre d'étapes le nombre d'étages du pipeline

Gain en performance

- Dans l'exemple précédant, la machine débute l'exécution d'une instruction à chaque cycle et le pipeline est pleinement occupé à partir du quatrième cycle
- Le gain obtenu dépend donc du nombre d'étages du pipeline
- Pour exécuter n instructions, chaque instruction s'exécute en k cycles d'horloge, il faut :n*k cycles d'horloge pour une exécution séquentielle
  - voir exemple: 4étages, 1cycle par étage, 3instructions :temps exécution)4 :cycles))3 *instr)=12 cycles,
- Avec un pipeline de k étages et k cycles d'horloge pour exécuter la première instruction puis n-1 cycles pour les n-1 instructions suivantes:
  - voir exemple: 4étages, 1cycle par étage, 3instructions :temps exécution)4 :cycles 1er instr.))1+cycle 2é instr.))1 +cycle 3é instr.) 6= cycles,
- [illegible]

Les limites

- La mise en place d'un pipeline pose plusieurs problèmes.
- plus le pipeline est long, plus le nombre de cas où il n'est pas possible d'atteindre la performance maximale est élevé.
- Les principaux cas où la performance d'un processeur pipeliné peut être dégradé:
  1. Les instructions ont besoins d'une même ressource de processeur)conflit de dépendance,(
  2. lorsqu'une instruction produit un résultat et que l'instruction suivante utilise ce résultat avant qu'il n'ait pu être écrit dans un registre,